

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-271389
 (43)Date of publication of application : 26.09.2003

(51)Int. Cl. G06F 9/445
 G06F 17/28

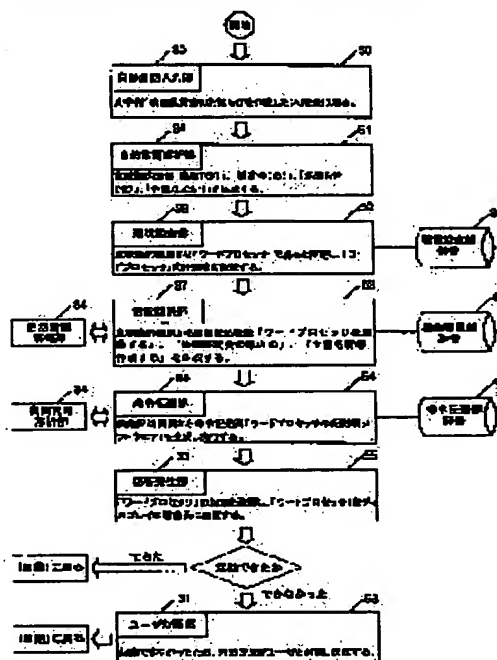
(21)Application number : 2002-076319 (71)Applicant : ARAKI SHUICHI
 OGURA MICHIO
 (22)Date of filing : 19.03.2002 (72)Inventor : ARAKI SHUICHI
 OGURA MICHIO

(54) METHOD FOR OPERATING SOFTWARE OBJECT IN NATURAL LANGUAGE AND ITS PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To realize a natural language interface having generality to integrally operate a plurality of different software/objects and flexibility to perform proper processing even at the time of receiving natural language expressions such as the request, desire, and purpose of a user as inputs.

SOLUTION: The inputted character column of natural language is defined as the expression of the request of a user, and the character string is analyzed, and an optimal software/object for performing processing corresponding to the request is selected, and function description expressions for making the software/object perform the processing is intermediately generated, and the function description expression is converted into a performable instruction description column by an OS or program.



LEGAL STATUS

[Date of request for examination] 19.10.2004

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

BEST AVAILABLE COPY

(12) 公開特許公報 (A)

(11)特許出願公開番号
特開2003-271389

(43)公開日 平成15年9月26日(2003.9.26)

(51) Int. Cl. ⁷	識別記号	F I	テーマコード (参考)
G06F 9/445		G06F 17/28	Z 5B076
17/28		9/06	650 A 5B091

審査請求 未請求 請求項の数2 OL (全11頁)

(71)出願人 502098204
荒木 修一
滋賀県大津市秋葉台3-8

(71)出願人 502098215
小倉 道雄
京都府京都市北区平野上柳町11-72

(72)発明者 荒木 修一
滋賀県大津市秋葉台 3-8

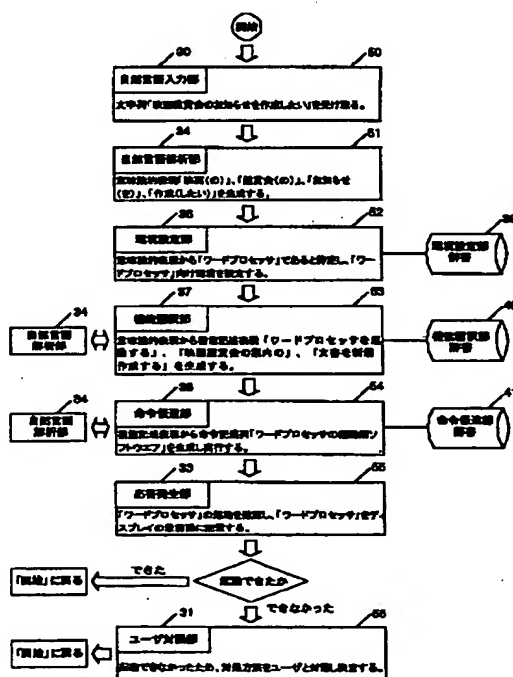
(72)発明者 小倉 道雄
京都市北区平野上柳町11-72

(74)代理人 100095670
弁理士 小林 良平

最終頁に続く

(57) 【要約】

【解決手段】 入力された自然言語の文字列をユーザの要求の表現ととらえて文字列の解析を行い、その要求に応じた処理を行うのに最適なソフトウェア・オブジェクトを選択し、そのソフトウェア・オブジェクトに前記処理を行わせるための機能記述表現を中間的に生成し、その機能記述表現をOS又はプログラムにより実行可能な命令記述列に変換する。



【特許請求の範囲】

【請求項1】 要求を表現した自然言語の文字列を所定の入力手段から受け取る手順、

前記文字列が示す言葉又は文の意味を解析して意味論的表現を生成する手順、

前記要求に応じた処理を行なうのに最適なソフトウェア・オブジェクトを前記意味論的表現に基づいて選択するとともに、該ソフトウェア・オブジェクトを操作するための環境を設定する手順、

前記意味論的表現を翻訳して、前記要求に応じた処理を前記ソフトウェア・オブジェクトに実行させるために該ソフトウェア・オブジェクトに与えるべき操作命令を正規化した言葉から成る機能記述表現を生成する手順、

前記機能記述表現から前記ソフトウェア・オブジェクトの実行可能な命令を生成して該ソフトウェア・オブジェクトに伝達する手順、及び、

前記命令を受けて前記ソフトウェア・オブジェクトが実行した処理の結果を前記ユーザが認識可能な所定の形態で出力する手順、を含む処理をコンピュータに実行させることを特徴とする、自然言語によるソフトウェア・オブジェクトの操作方法。

【請求項2】 請求項1に記載の方法に従った処理をコンピュータに実行させるためのプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータ上で動作するソフトウェア・オブジェクトを自然言語で操作する方法及びそのためのプログラムに関する。本明細書においてソフトウェア・オブジェクトとは、パーソナル・コンピュータやマイコン制御機器等の電子機器を制御するためのオペレーティング・システム（OS）や、OS上で動作するアプリケーション・プログラムのことをいう。また、入力機器（キーボード、マイク、手書きタブレット等）から受け取った信号を処理して自然言語の文字列を生成し、その文字列を解析し、解析結果に基づいてソフトウェア・オブジェクトの操作命令を生成するようなシステムのことを本明細書では自然言語インターフェースと呼ぶ。

【0002】

【従来の技術】ソフトウェア・オブジェクトを自然言語で操作するための自然言語インターフェースに関する研究は従来より盛んに行われている。例として、特開平8-147096号公報に記載の手書き入力方法及び装置、特開平6-75692号公報に記載の情報処理装置、特開平6-131108号公報に記載の情報入力装置、特開平6-282566号公報に記載の情報入力装置等が挙げられる。これら従来の自然言語インターフェースは、ソフトウェア・オブジェクトに用意された内部機能を自然言語で呼び出すために用いられる。例えば、特開平6-75692には、ユーザが「大きくする」と手書き入力すると予め指定された文字列が倍

角文字になるようなワードプロセッサが開示されている。また、特開平8-147096には、ユーザが「録画」と手書き入力すると録画処理が始まるような制御系を有するビデオデッキが開示されている。

【0003】

【発明が解決しようとする課題】従来の自然言語インターフェースは、例えばワードプロセッサ・プログラムやビデオデッキの制御プログラムといった個々のソフトウェア・オブジェクトに特化したものばかりであって、あるソフトウェア・オブジェクトのために開発された自然言語インターフェースを他のソフトウェア・オブジェクトの操作に転用することは基本的に想定されていない。このため、ソフトウェア開発者は、あるソフトウェア・オブジェクト用の自然言語インターフェースを装備したい場合、専用の自然言語インターフェースを新たに開発するために相当な労力を費やさなければならない。

【0004】また、従来の自然言語インターフェースでは、ソフトウェア・オブジェクトに予め用意された内部機能呼び出すための命令が入力されることが想定されている。従って、ユーザは、ソフトウェア・オブジェクトがどのような機能を備えており、それをどのような自然言語で呼び出せばよいかに関する情報（知識）を予め持ち合わせていなければならない。つまり、ユーザの要求に応える形でソフトウェア・オブジェクトが機能するというより、ソフトウェア・オブジェクトの機能に合わせてユーザが命令を出すのである。このような形態に留まる限り、自然言語によるソフトウェア・オブジェクトの操作は柔軟性を欠いたものとならざるを得ない。例えば、ユーザが「映画鑑賞会のお知らせを作成したい」と考え、その考えをそのまま言葉にして入力したとする。「映画鑑賞会のお知らせを作成したい」は、いわばユーザの要求、願望又は目的を表現したものであって、ソフトウェア・オブジェクトの何らかの機能を明示的に呼び出すための命令ではない。従来の自然言語インターフェースではこのような入力に対して適切な処理を実行することは不可能である。

【0005】本発明は以上のような課題を解決するために成されたものであり、その目的とするところは、複数の異なるソフトウェア・オブジェクトを統一的に操作できる汎用性と、ユーザの要求、願望、目的等の自然言語表現を入力として受け取ったときでも適切な処理ができる柔軟性を備えた自然言語インターフェースを実現するための技術を提供することにある。

【0006】

【課題を解決するための手段】上記課題を解決するために成された本発明に係る自然言語によるソフトウェア・オブジェクトの操作方法は、要求を表現した自然言語の文字列を所定の入力手段から受け取る手順、前記文字列が示す言葉又は文の意味を解析して意味論的表現を生成する手順、前記要求に応じた処理を行なうのに最適なソ

ソフトウェア・オブジェクトを前記意味論的表現に基づいて選択するとともに、該ソフトウェア・オブジェクトを操作するための環境を設定する手順、前記意味論的表現を翻訳して、前記要求に応じた処理を前記ソフトウェア・オブジェクトに実行させるために該ソフトウェア・オブジェクトに与えるべき操作命令を正規化した言葉から成る機能記述表現を生成する手順、前記機能記述表現から前記ソフトウェア・オブジェクトの実行可能な命令を生成して該ソフトウェア・オブジェクトに伝達する手順、及び、前記命令を受けて前記ソフトウェア・オブジェクトが実行した処理の結果を前記ユーザが認識可能な所定の形態で出力する手順、を含む処理をコンピュータに実行させることを特徴とする。

【0007】また、本発明は、コンピュータに上記のような処理を実行させるためのプログラムを提供する。

【0008】

【発明の実施の形態】本発明による処理の手順について図面を参照しながら具体的に説明する。

【0009】まず、ユーザがコンピュータの入力手段を用いて自然言語で入力する文字列を受け取る（ステップ50）。入力手段は、キーボード、手書き入力装置、音声入力装置等のハードウェアと、そのハードウェアの出力信号から自然言語の文字列を生成するソフトウェア（キーボードドライバ、パターン認識ソフトウェア、音声認識ソフトウェア等）により構成する。ここでは「映画鑑賞会のお知らせを作成したい」という文字列が入力されたものとする。

【0010】次に、上記のようにして生成された文字列を解析して意味論的表現を生成する（ステップ51）。この処理は、形態素解析、構文解析、意味解析等の手順を含む周知の自然言語処理方法で実行することができる。ここでは、「映画（の）」「鑑賞会（の）」「お知らせ（を）」「作成（したい）」という意味論的表現が生成されたものとする。

【0011】次に、ユーザの要求に応じた処理を行なうのに最適なソフトウェア・オブジェクトを上記意味論的表現に基づいて選択する（ステップ52）。ソフトウェア・オブジェクトの選択は、意味論的表現とソフトウェア・オブジェクトとを関連づけた辞書（以下、環境設定部辞書と呼ぶ）を用いて行われる。環境設定部辞書の一例を図2に示す。図2の辞書を用いた場合、「映画（の）」「鑑賞会（の）」「お知らせ（を）」「作成（したい）」という意味論的表現から、ワードプロセッサの評価点=1.7電子メールクライアントの評価点=0.2作図ソフトの評価点=0.2というようにソフトウェア・オブジェクト毎の評価点が得られ、最も評価点の高い「ワードプロセッサ」が最適なソフトウェア・オブジェクトとして選択される。ここでは、最も評価点の高いソフトウェア・オブジェクトを自動的に選択してもよいが、ユーザから承認を得るステップを経た後でソフトウ

ウェア・オブジェクトを選択するようにしてもよい。

【0012】次に、上記のように選択したソフトウェア・オブジェクトを操作するための環境を設定する（ステップ53）。具体的には、ソフトウェア・オブジェクトの機能を正規化された言葉に翻訳するための辞書（以下、機能翻訳部辞書と呼ぶ）を用いて、意味論的表現を機能記述表現に翻訳する。機能翻訳部辞書の一例を図3に示す。図3の機能翻訳部辞書50は、入力語と置き換え可能な出力語（訳語）を一つの変換対とした変換表である。この変換表により、例えば入力語である「作成」は、「作る」という出力語に変換できる。図3の例ではさらに、一つの入力語に対して複数の出力語の候補がある場合に、適切な出力語を選択するための付加情報として、入力語の種類と、出力語への変換の妥当性を示す評価点を各変換対に与えている。この評価点は、作業の過程で動的に変化する。

【0013】図3の辞書に基づく入力語から出力語への変換（翻訳）手順を更に具体的に説明する。例えば「作成」という言葉を見ると、まずこの言葉は「作る」に翻訳される。「作る」は更に「文書を作る」、「図形を作る」及び「メールを作る」のいずれかに翻訳可能であるが、ここではワードプロセッサがソフトウェア・オブジェクトとして選択されているから、「文書を作る」の評価点が最も高くなっている。従って、「文書を作る」が自動的に（あるいはユーザから承認を得た後で）訳語（訳文）として選択される。「文書を作る」は更に「ワードプロセッサを起動する／文書を新規作成する」に翻訳される。「ワードプロセッサを起動する／文書を新規作成する」に対応するエントリは辞書に存在しない。従って、「ワードプロセッサを起動する／文書を新規作成する」が「作成」の機能技術表現となる。同様に、「映画（の）」「鑑賞会（の）」「お知らせ（を）」「作成（したい）」という意味論的表現の各語を再帰的に翻訳すると、最終的に、「ワードプロセッサを起動する」「映画鑑賞会の案内の」「文書を新規作成する」という機能記述表現が得られる。

【0014】次に、上記機能記述表現からソフトウェア・オブジェクトの操作命令を生成し、その命令を実行する（ステップ54）。例えば、「ワードプロセッサを起動する」という機能記述表現に対しては、ハードディスクの所定場所に保存されたワードプロセッサのプログラムを読み出して実行するための命令記述列を生成し、OSに実行させる。また、「文書を新規作成する」という機能記述表現に対しては、文書新規作成機能呼び出すための命令記述列を生成し、OSを通じてワードプロセッサのプログラムに実行させる。OSに渡すべき命令記述列はOSのアプリケーション・プログラミング・インターフェース（API）の仕様に従って作成し、ワードプロセッサのプログラムに渡すべき命令記述列はワードプロセッサのAPIの仕様に従って作成する。命令記述

列の例としては、プログラムを起動するためのコマンドラインや、動作中のプログラム環境の内部で各種機能を利用するためのスクリプトが挙げられる。

【0015】次に、OS又はソフトウェア・オブジェクトにより命令記述列を実行して得られた結果をユーザの認識可能な所定の形態で出力する。例えば、「ワードプロセッサを起動する」に対応する命令が正常に実行できた場合、ワードプロセッサのウィンドウがコンピュータの画面上で最前面に表示される(ステップ55)。また、「文書を新規作成する」に対応する命令が正常に実行できた場合、ワードプロセッサのウィンドウ内に白紙の文書が生成される。なお、操作が正常に実行できなかった場合、所定のエラー処理が実行される(ステップ56)。

【0016】

【発明の効果】以上のように、本発明は、自然言語で入力されたユーザの要求に応じた処理を行うのに最適なソフトウェア・オブジェクトを自動的に選択し、更にそのソフトウェア・オブジェクトを操作するための適切な命令記述列を自動的に生成する基本的なアーキテクチャを提供するものである。このような本発明により、ソフトウェア・オブジェクトと自然言語インターフェースとの連携を従来よりも簡単に実現することができる。すなわち、ソフトウェア・オブジェクトを操作するための命令記述列を定めるとともに、各命令記述列と機能記述表現とを関連づける辞書データを作成するだけで、そのソフトウェア・オブジェクトを自然言語で操作する仕組みを実現することができる。

【0017】また、従来の技術では、入力された自然言語の文字列をユーザの命令(すなわち、本発明の機能記述表現に相当するもの)ととらえるのに対し、本発明では、入力された自然言語の文字列をユーザの要求の表現ととらえて文字列の解析を行い、各種辞書を用いてソフトウェア・オブジェクトの機能記述表現を中間的に生成する。言い換えると、従来はソフトウェア・オブジェクトの機能をどのように利用したいかをユーザに言葉で表現させていたのに対し、本発明では自分が何をしたいかをユーザに言葉で表現させる。従って、ユーザは、どのようなソフトウェア・オブジェクトが利用できるかと、各ソフトウェア・オブジェクトにどのような機能が備わっているかといったことを事前に知らなくても、自分が何をしたいかを直接言葉で表現するだけで、ソフトウェア・オブジェクトを操作することができる。

【0018】

【実施例】本発明に従って構成された自然言語インターフェースを備えるコンピュータ・システムの一例の概略構成を図4に示す。このコンピュータ・システムは、一般のパーソナル・コンピュータを利用して構成されたものであって、CPU(中央演算処理装置)10、ROM(読出し専用主記憶装置)11、RAM(読書き可能主

記憶装置)12、外部記憶装置コントローラ13と外部記憶装置(補助記憶装置)14、外部と連携するためのネットワーク・コントローラ15、ユーザ・インターフェース・アダプター16、ディスプレイ・コントローラ21とディスプレイ22を備えている。ユーザ・インターフェース・アダプター16には、言葉列を入力するための各種入力機器(キーボード17、音声入力用マイク18、マウス19、手書き入力用タブレット20)が接続されている。

【0019】本実施例のシステムの機能的構成を図5に示す。図5において、自然言語入力部30は、自然言語による言葉、言葉列又は文(以下、これらをまとめて「言葉」と呼ぶ)の入力を受け付け、その言葉を表す文字列を生成する手段である。言葉の入力方法としては、キーボード17を用いたキー入力、マイク18を用いた音声入力、マウス19を用いた画面上の文字入力パネルの操作、あるいは、タブレット20を用いた手書き入力から選択できる。もちろん、これら以外の方法でも、そのための入力機器とそのためのソフトウェア(ドライバ)を用意さえすれば、言葉の入力に利用できる。

【0020】自然言語解析部34は、自然言語解析、辞書によるパーシング、対話形式構文生成、カテゴリ辞書マネージャー機能を有しており、上記文字列を解析して意味論的表現を生成する。文字列の解析には、自然言語処理の分野で一般的に知られた技術が利用できる。例えば、奈良先端科学技術大学院大学の「茶釜」、京都大学の「KNP」等の自然言語解析エンジンが既に知られているが、このような既存のエンジンを用いて自然言語解析部34を構成することができる。

【0021】環境設定部36は、意味論的表現に現れる全ての概念で環境設定部辞書39(図2参照)を検索し、ユーザの要求に応じた処理を行うのに最適なソフトウェア・オブジェクトを選択し、そのソフトウェア・オブジェクトの操作環境を設定する。環境設定部辞書39には、意味論的表現に用いられる概念とシステムで利用可能なソフトウェア・オブジェクトとを関連づける情報や、ソフトウェア・オブジェクトの環境設定方法に関する情報が保存されている。環境設定には、以降の処理に利用する辞書の設定、ソフトウェア・オブジェクトが動作する機器内の環境の設定が含まれる。環境設定方法が自然言語で記述されている場合、自然言語解析部34を介して一連の動作を再帰的に繰り返す。

【0022】機能翻訳部37は、意味論的表現に現れる全ての概念で機能翻訳部辞書40(図3参照)を検索し、辞書に記憶されているソフトウェア・オブジェクトの機能に合った機能記述表現に置き換える。辞書に自然言語そのものが登録されている可能性があるため、この置換処理は自然言語解析部34を介した再帰的処理となる。最終的には生成される機能記述表現は、正規化された言葉から成る意味論的表現になる。なお、辞書内に未

定義部分が定義されていた場合、機能翻訳部 3 7 はユーザ対話部 3 1 を介して未定義部分の指定をユーザから受け取る。

【0023】命令伝達部 3 8 は、機能翻訳部 3 7 により作成された機能記述表現に現れる全ての概念で命令伝達部辞書 4 1 を検索し、辞書に記憶されているソフトウェア・オブジェクト 4 2 の機能を実行するための命令記述列を生成する。命令記述列は、例えば、該当ソフトウェア・オブジェクト 4 2 の API とパラメータ、あるいは、コマンドストリームを介して渡すコマンド列である。命令伝達部 3 8 は命令記述列を実行し、該当ソフトウェア・オブジェクト 4 2 の機能を実行する。

【0024】応答発生部 3 3 は、命令伝達部 3 8 により実行されたソフトウェア・オブジェクト 4 2 の実行結果を受け取り、ユーザが所望する形態で応答する。応答の形態は、例えば、ディスプレイ 2 2 への表示、プリンタ（図示せず）による印刷、データベースへの情報格納、機器の制御等、様々なものが考えられる。ソフトウェア・オブジェクト 4 2 の機能を実行して得られた結果が不十分なものであり、ユーザの所望した形態で応答できない場合、応答発生部 3 3 はユーザ対話部 3 1 を介してユーザにメッセージを提示し、必要があればユーザに指示を求める。

【0025】辞書管理部 3 5 は、環境設定部辞書 3 9、機能翻訳部辞書 4 0、命令伝達部辞書 4 1 の新規情報の作成、格納情報の変更、削除、閲覧表示を行う。さらに、制御部 4 2 は自然言語入力部 3 0、自然言語解析部 3 4、環境設定部 3 6、機能翻訳部 3 7、命令伝達部 3 8、応答発生部 3 3、ユーザ対話部 3 1、辞書管理部 3 5 に必要なデータを受け渡し、一連の動作を制御する。

【0026】「映画鑑賞会のお知らせを作成したい」という入力文字列の処理を本実施例のシステムで行う手順について図 1 ～ 図 3 を参照しながら説明する。

【0027】映画鑑賞会のお知らせを作成しよう考えたユーザが、例えばキーボード 2 4 を用いて「映画鑑賞会のお知らせを作成したい」という文を打ち込むと、自然言語入力部 3 0 が、キーボード入力インターフェースを通じて「映画鑑賞会のお知らせを作成したい」という文字列を受け取る（ステップ 5 0）。この文字列は自然言語解析部 3 4 に渡される。

【0028】自然言語解析部 3 4 は受け取った文字列を解析し、例えば「映画（の）」、「鑑賞会（の）」、「お知らせ（を）」及び「作成（したい）」という、構文および意味的に分離した 4 つの言葉から成る意味論的表現を生成する（ステップ 5 1）。この意味論的表現は環境設定部 3 6 に渡される。

【0029】環境設定部 3 6 は、環境設定部辞書 3 9（図 2 参照）に基づいて上記 4 つの言葉のソフトウェア・オブジェクト別評価点を取得し、総合的な評価点の最も高いソフトウェア・オブジェクトは「ワードプロセッ

サ」であると判断し、環境設定部辞書 3 9 に格納されている「ワードプロセッサ」向けの環境設定処理を実行する（ステップ 5 2）。環境設定処理には、機能翻訳部辞書 4 0 と命令伝達部辞書 4 1 の設定とコンピュータ資源の確認及び確保が含まれる。

【0030】機能翻訳部 3 7 は、機能翻訳部辞書 4 0（図 3 参照）に基づいて上記 4 つの言葉をソフトウェア・オブジェクトが提供する機能およびその組み合わせで置き換えることにより、意味論的表現を機能記述表現に翻訳する（ステップ 5 3）。例えば、「作る」については「文書を作る」、「図形を作る」の二つの出力語（訳語）の候補があるが、ここでは「文書を作る」の評価点が最も高いので、「文書を作る」に変換する。こうして機能翻訳部 3 7 は、自然言語解析部 3 4 が生成した「映画（の）」、「鑑賞会（の）」、「お知らせ（を）」、「作成（したい）」という意味論的表現を図 3 の機能翻訳部辞書 4 0 で再帰的に検索・置換し、「ワードプロセッサを起動する」、「映画鑑賞会の案内の」、「文書を新規作成する」という機能記述表現を生成する。再帰的検索・置換時には、自然言語解析部 3 4 を利用して意味論的表現を動的に変更する。

【0031】次に、命令伝達部 3 8 が命令伝達部辞書 4 1 を用いて命令記述列を生成する（ステップ 5 4）。

「ワードプロセッサを起動する」を例にとると、まず、自然言語解析部 3 4 がこの文字列を解析し「ワードプロセッサ（を）」、「起動（する）」を分離する。次に、命令伝達部 3 8 がこれらの概念で命令伝達部辞書 4 1 を検索し命令記述列を生成する。この場合、「起動（する）」が、特定のワードプロセッサ用アプリケーションをオペレーティング・システムが提供する API を介して起動する実行可能ソフトウェアに置き換わり、命令伝達部 3 8 が実行する。命令記述列の生成も再帰的に検索・置換を行い、自然言語解析部 3 4 を利用して意味論的表現を動的に変更する。

【0032】次に、応答生成部 3 3 はワードプロセッサが起動したことを確認し、ディスプレイの最前面にワードプロセッサを配置する（ステップ 5 5）。なお、何らかの障害によりワードプロセッサが起動しなかった場合、応答生成部 3 3 は、ユーザ対話部 3 1 を介してユーザと対話し、対処方法を決定する（ステップ 5 6）。ワードプロセッサが起動したら、ユーザは引き続き自分のやりたいこと（要求）を言葉で順次入力することによりドキュメントを作成する。例えば、入力される言葉は、例えば「タイトルを映画鑑賞会のお知らせにする」、「タイトルを強調したい」といったものである。なお、「終了する」という言葉を入力するとプログラムが終了する。

【0033】前記例では、ユーザが一連の自然言語入力を行うことで「映画鑑賞会のお知らせ」を作成した。次に、この「お知らせ」を作成する操作手順をシステムに

登録することにより同様の「お知らせ」を簡単に再作成できるようにする手順について説明する。ここでは例として、日時、場所、映画名、映画の紹介を自由に変更した「お知らせ」を作成できるようにすることを考える。

【0034】まず、辞書管理部35を介して、先に説明したような一連の操作に相当する機能記述表現を機能翻訳部辞書40に適宜名前（ここでは「映画鑑賞会のお知らせ」とする）で登録する。次に、辞書管理部35を介して、機能翻訳部辞書40に登録された前記一連の機能記述表現に含まれる文字列のうち、日時、場所、映画名、映画の紹介に相当する部分を未定義部分として再定義する（ステップ60）。

【0035】次に、辞書管理部35を介して、「映画鑑賞会のお知らせ」という文字列をワードプロセッサ・ソフトウェア・オブジェクトと関連づけて環境設定部辞書39に登録する（ステップ61）。

【0036】上記のように機能翻訳部辞書40及び環境設定部辞書39に「映画鑑賞会のお知らせ」のエントリを追加した後、ユーザが自然言語入力部30を介して「映画鑑賞会のお知らせを作成する」と自然言語入力を行なうと、自然言語解析部34と環境設定部36が図1の例と同様の処理を行ない、意味論的表現を生成する（ステップ62）。

【0037】次に、機能翻訳部37は、前記意味論的表現を機能記述表現に翻訳する際、「映画鑑賞会のお知らせ」を先に機能翻訳部辞書40に登録した一連の機能記述表現に置換え（ステップ63）、その機能記述表現に対して先に説明したような再帰的翻訳処理を実行する（ステップ64）。機能記述表現に含まれる未定義部分（日時、場所、映画名、映画の紹介）を見つけると（ステップ65）、機能翻訳部37は、ユーザ対話部31を介してユーザにその部分の定義を問う。ユーザが定義に相当する言葉（文字列）を入力すると、機能翻訳部37は未定義部分をその言葉で置き換える（ステップ66）。このようにして、ユーザはユーザ対話部31のガイダンスに沿い、日時、場所、映画名、映画の紹介を入力することで、映画鑑賞会のお知らせを簡単に作成できる。

【0038】以上、本発明の実施例について説明したが、実施例は上記に限られるものではない。例えば、上記実施例ではパーソナル・コンピュータにインストールされた複数のアプリケーション・ソフトウェア・オブジェクトを自然言語インターフェースを通じて操作するものとしたが、LANやインターネット等のネットワークに接続された複数のネットワーク対応電子機器（コンピュータも含む）を、同ネットワークに接続された制御機器の自然言語インターフェースを通じて操作できるようにすることも可能である。従って、例えば、家庭内のLANに接続されたネットワーク対応型電化製品に対して一つの音声入力型コントローラを設けるというシステムを実現することもできる。

【図面の簡単な説明】

【図1】 本発明に係る方法によるソフトウェア・オブジェクトの操作手順の一例を示すフローチャート。

【図2】 環境設定部辞書の構成例。

【図3】 機能翻訳部辞書の構成例。

【図4】 本発明の一実施例であるコンピュータ・システムのハードウェア構成を示すブロック図。

【図5】 本発明に従って構成された自然言語インターフェースの機能的構成を示すブロック図。

【図6】 本発明に係る方法によるソフトウェア・オブジェクトの操作手順の別の例を示すフローチャート。

【符号の説明】

30…自然言語入力部

31…ユーザ対話部

32…制御部

33…応答発生部

34…自然言語解析部

35…辞書管理部

36…環境設定部

37…機能翻訳部

38…命令伝達部

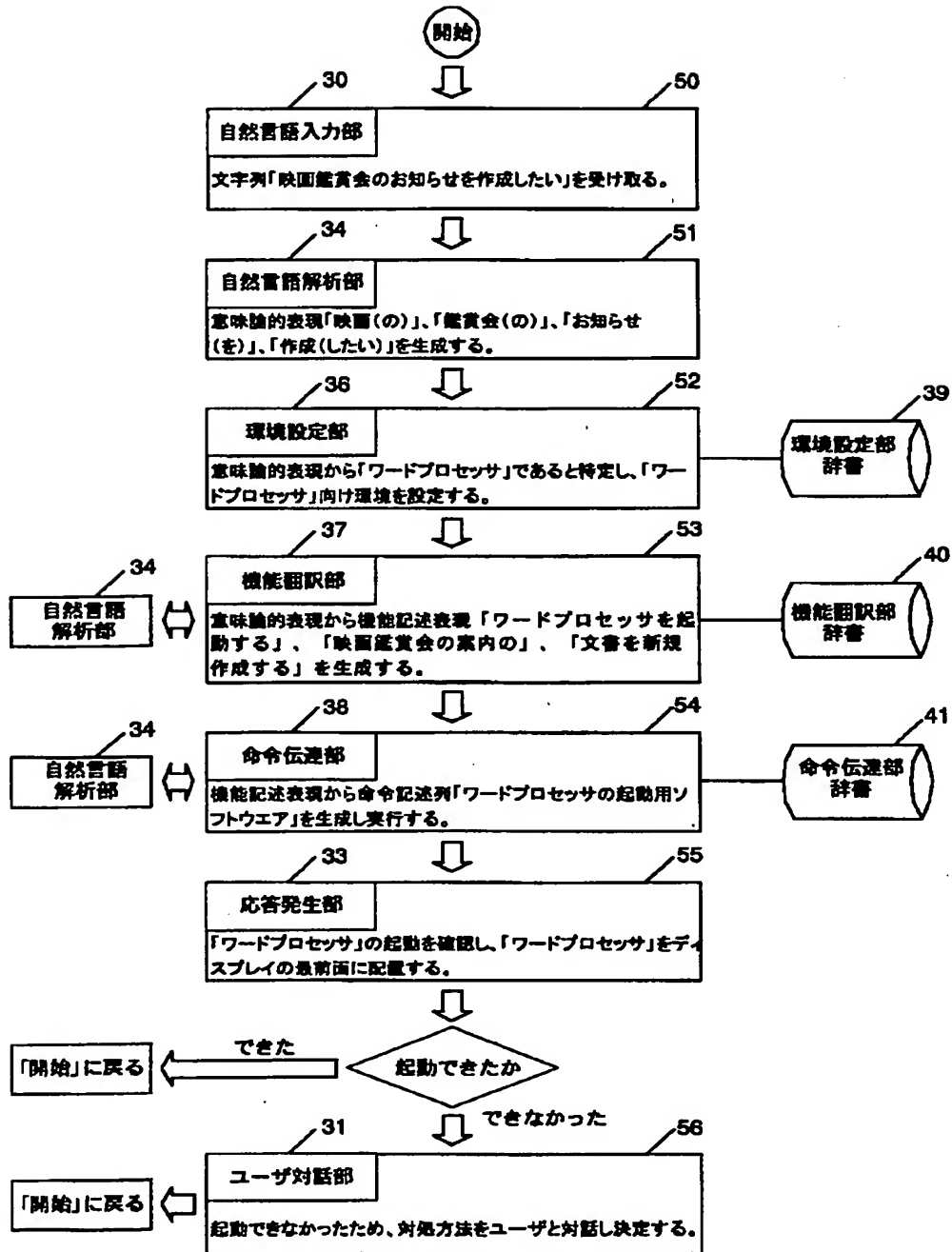
39…環境設定部辞書

40…機能翻訳部辞書

41…命令伝達部辞書

42…ソフトウェア・オブジェクト

【図 1】



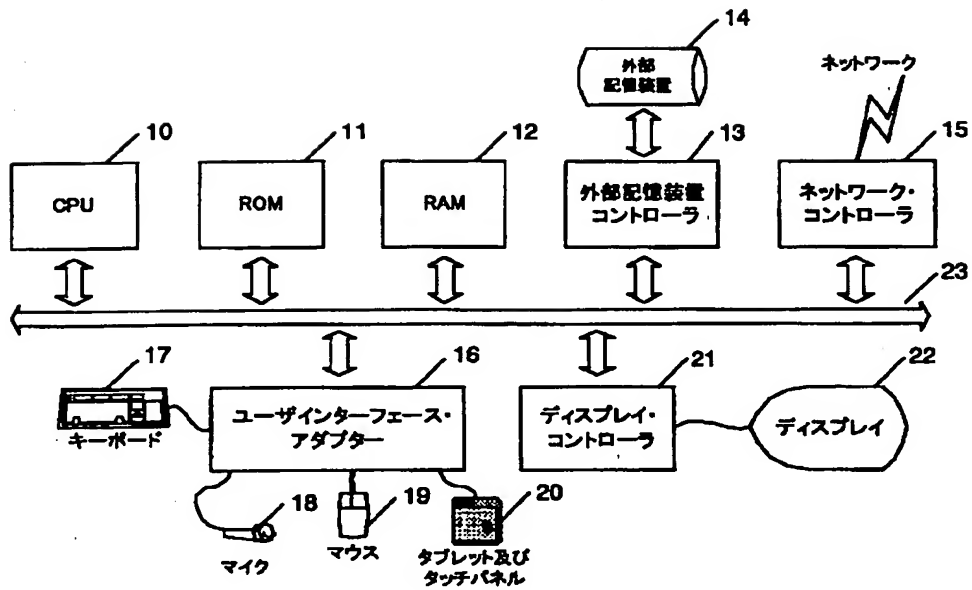
【図2】

入力値	属性	重み値	ソフトウェア・ツールの名称
：			
お知らせ	名詞	1	ワードプロセッサ
お知らせ	名詞	1	メール
：			
案内	名詞	0.8	ワードプロセッサ
案内	名詞	0.4	メール
：			
作成	動詞	0.7	ワードプロセッサ
作成	動詞	0.1	図形作成ツール
作成	動詞	0.2	メール
：			
送る	動詞	1	メール
：			

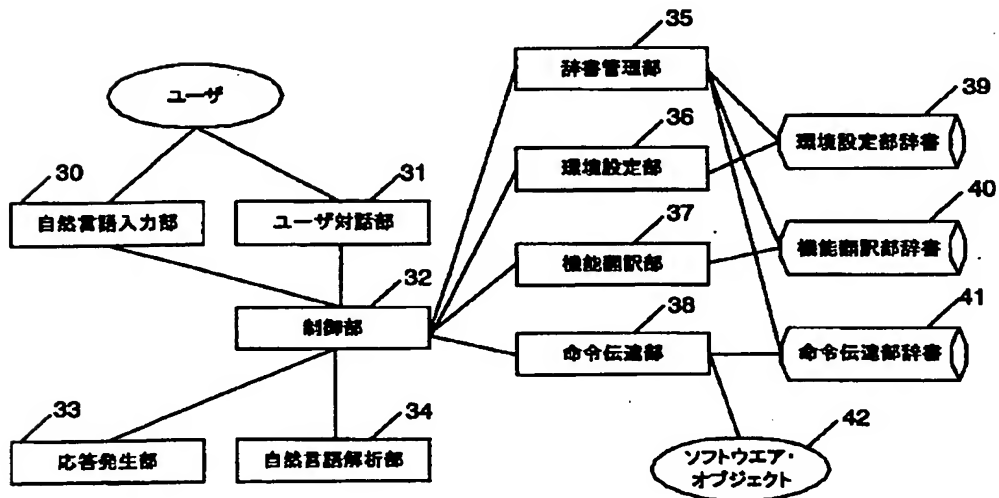
【図3】

入力値	属性	重み値	ソフトウェア・ツールの名称
：			
作成	動詞	1	作る
：			
製作	動詞	1	作る
：			
お知らせ	名詞	1	案内
：			
ご案内	名詞	1	案内
：			
案内	名詞	0.6	案内の文書
案内	名詞	0.4	案内のメール
：			
作る	動詞	0.7	文書を作る
作る	動詞	0.1	図形を作る
作る	動詞	0.2	メールを作る
：			
送る	動詞	1	メールを送る
：			
文書を作る	動詞	1	ワードプロセッサを起動する/文書を新規作成する
：			
図形を作る	動詞	1	図形作成ツールを起動する/図形を新規作成する
：			
メールを作る	動詞	1	メールツールを起動する/メールを新規作成する
メールを送る	動詞	1	メールを作る/メールを送信する
：			

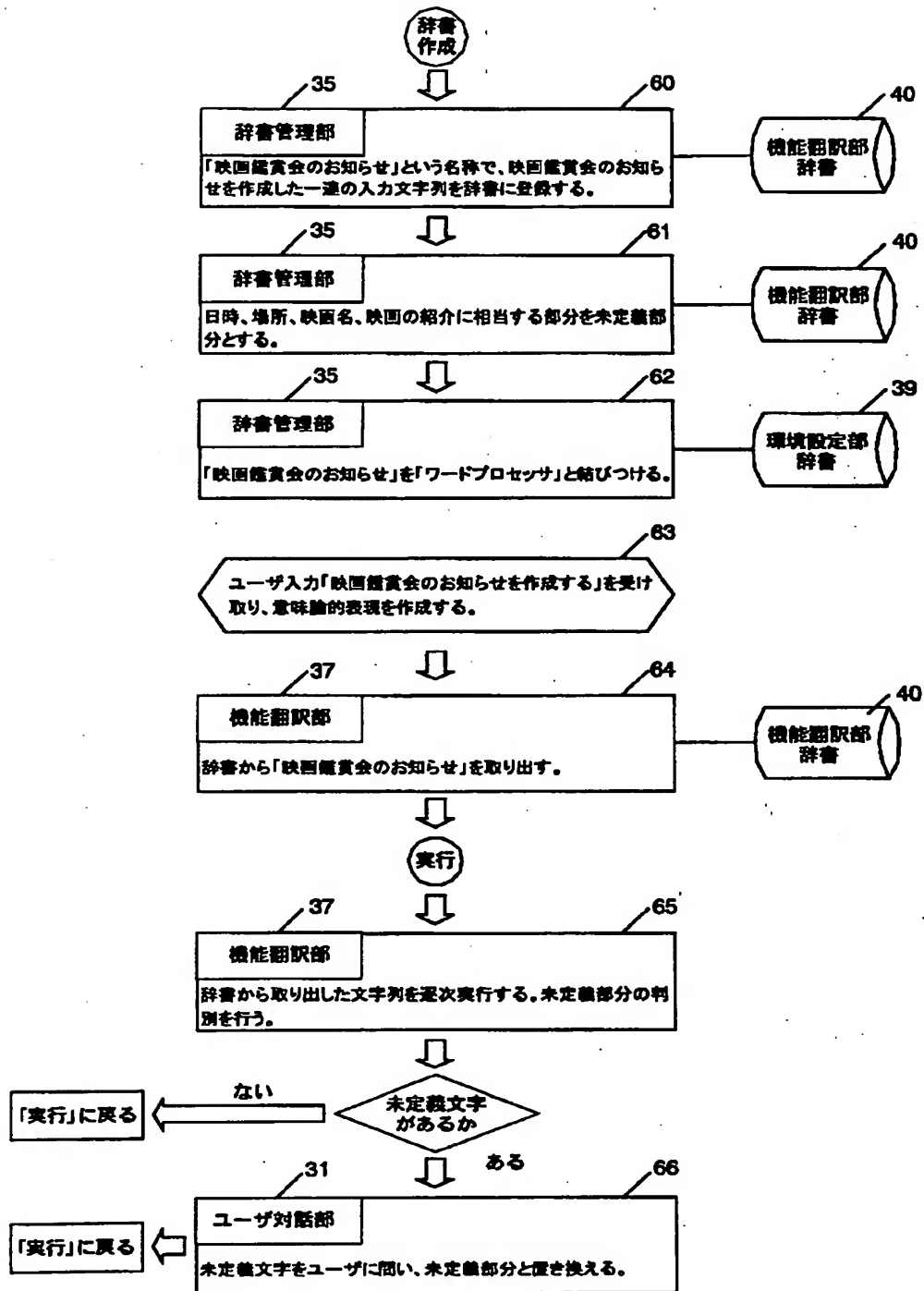
【図4】



【図5】



【図6】



フロントページの続き

Fターム(参考) 5B076 AA03 AA12 AA14 AB17 DA03
DC04 DC08
5B091 AA15 DA00 EA24

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-271389

(43)Date of publication of application : 26.09.2003

(51)Int.Cl.

G06F 9/445
G06F 17/28

(21)Application number : 2002-076319

(71)Applicant : ARAKI SHUICHI
OGURA MICHIO

(22)Date of filing : 19.03.2002

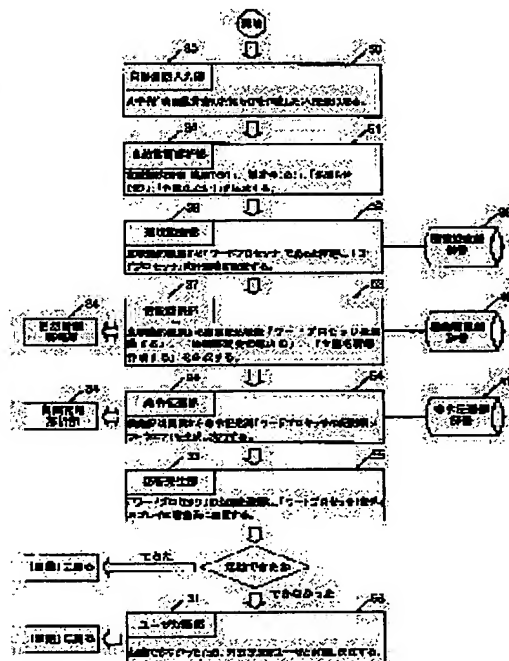
(72)Inventor : ARAKI SHUICHI
OGURA MICHIO

(54) METHOD FOR OPERATING SOFTWARE OBJECT IN NATURAL LANGUAGE AND ITS PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To realize a natural language interface having generality to integrally operate a plurality of different software/objects and flexibility to perform proper processing even at the time of receiving natural language expressions such as the request, desire, and purpose of a user as inputs.

SOLUTION: The inputted character column of natural language is defined as the expression of the request of a user, and the character string is analyzed, and an optimal software/object for performing processing corresponding to the request is selected, and function description expressions for making the software/object perform the processing is intermediately generated, and the function description expression is converted into a performable instruction description column by an OS or program.



LEGAL STATUS

[Date of request for examination]

19.10.2004

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

*** NOTICES ***

JPO and NCIPI are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The procedure of receiving the character string of the natural language expressing a demand from a predetermined input means, While choosing the optimal software object for performing the procedure which analyzes the semantics of the language which said character string shows, or a sentence, and generates semantic representation, and processing according to said demand based on said semantic representation The procedure of setting up the environment for operating this software object, and said semantic representation are translated. The procedure which generates the functional description expression which consists of the language which normalized the operating instructions which should be given to this software object in order to make said software object perform processing according to said demand, The procedure which generates the instruction which can perform said software object from said functional description expression, and is transmitted to this software object, And operating instructions of the software object by natural language characterized by making a computer perform processing including the procedure which outputs the result of the processing which said software object performed in response to said instruction with the predetermined gestalt which said user can recognize.

[Claim 2] The program for making a computer perform processing according to an approach according to claim 1.

[Translation done.]

*** NOTICES ***

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the method of operating the software object which operates on a computer with natural language, and the program for it. In this specification, a software object means the operating system (OS) for controlling electronic equipment, such as a personal computer and a microcomputer control equipment, and the thing of an application program which operates on OS. Moreover, the signal received from input devices (a keyboard, a microphone, handwriting tablet, etc.) is processed, the character string of natural language is generated, the character string is analyzed, and the thing of a system which generates the operating instructions of a software object based on an analysis result is called a natural language interface on these specifications.

[0002]

[Description of the Prior Art] Research on the natural language interface for operating a software object with natural language is done more briskly than before. As an example, the handwriting input approach given in JP,8-147096,A and equipment, an information processor given in JP,6-75692,A, an information input unit given in JP,6-131108,A, an information input unit given in JP,6-282566,A, etc. are mentioned. The natural language interface of these former is used in order to call the internal function prepared for the software object with natural language. For example, a word processor with which the character string beforehand specified that a user does a handwriting input to "It enlarges" becomes a double width character is indicated by JP,6-75692,A. Moreover, if a user does a handwriting input with an "image transcription", the videocassette recorder which has a control system from which image transcription processing begins is indicated by JP,8-147096,A.

[0003]

[Problem(s) to be Solved by the Invention] The conventional natural language interface is only what specialized in each software objects, such as for example, a word processor program and a control program of a videocassette recorder, and diverting the natural language interface developed for a certain software object to actuation of other software objects is not assumed fundamentally. For this reason, a software-development person has to spend a considerable effort to equip the natural language interface for some software objects, in order to newly develop the natural language interface of dedication.

[0004] Moreover, in the conventional natural language interface, it is assumed that the instruction for calling the internal function beforehand prepared for the software object is inputted. Therefore, the software object is equipped with what kind of function, and the user has to have beforehand the information (knowledge) about with what kind of natural language it should be called with him. That is, to compensate for the function of a software object, a user issues an instruction rather than a software object functions in the form which meets the demand of a user. As long as it stops at such a gestalt, actuation of the software object by natural language cannot but become a thing lacking in flexibility. For example, suppose that the user thought "I want to create information of a movie appreciation meeting", and inputted by making the idea into language as it is. It is not the instruction for "I want to create information of a movie appreciation meeting", expressing a demand of a user, a wish, or the purpose so

to speak, and calling a certain function of a software object clearly. It is impossible to perform suitable processing to such an input in the conventional natural language interface.

[0005] The place which accomplishes this invention in order to solve the above technical problems, and is made into the purpose is to offer the technique for realizing the natural language interface equipped with the flexibility which can perform suitable processing, even when the natural language expression of a demand of a user, a wish, the purpose, etc. is thought as an input to be the versatility which can operate systematically the software object from which plurality differs.

[0006]

[Means for Solving the Problem] The operating instructions of the software object by the natural language concerning this invention accomplished in order to solve the above-mentioned technical problem The procedure of receiving the character string of the natural language expressing a demand from a predetermined input means, While choosing the optimal software object for performing the procedure which analyzes the semantics of the language which said character string shows, or a sentence, and generates semantic representation, and processing according to said demand based on said semantic representation The procedure of setting up the environment for operating this software object, and said semantic representation are translated. The procedure which generates the functional description expression which consists of the language which normalized the operating instructions which should be given to this software object in order to make said software object perform processing according to said demand, The procedure which generates the instruction which can perform said software object from said functional description expression, and is transmitted to this software object, And it is characterized by making a computer perform processing including the procedure which outputs the result of the processing which said software object performed in response to said instruction with the predetermined gestalt which said user can recognize.

[0007] Moreover, this invention offers the program for making a computer perform the above processings.

[0008]

[Embodiment of the Invention] It explains concretely, referring to a drawing about the procedure of processing by this invention.

[0009] First, the character string which a user inputs with natural language using the input means of a computer is received (step 50). Hardware, such as a keyboard, a handwriting input unit, and an audio input unit, and the software (a keyboard driver, pattern recognition software, speech recognition software, etc.) which generates the character string of natural language from the output signal of the hardware constitute an input means. Here, the character string of "wanting to create information of a movie appreciation meeting" should be inputted.

[0010] Next, the character string generated as mentioned above is analyzed, and semantic representation is generated (step 51). This processing can be performed by the natural-language-processing approach of common knowledge including procedures, such as morphological analysis, syntax analysis, and a semantic analysis. here, the semantic representation "a movie (that)", "an appreciation meeting (that)", "information (**)", and "creation (I want to carry out)" should be generated

[0011] Next, the optimal software object for performing processing according to a demand of a user is chosen based on the above-mentioned semantic representation (step 52). Selection of a software object is performed using the dictionary (it is hereafter called a configuration section dictionary) which associated semantic representation and a software object. An example of a configuration section dictionary is shown in drawing 2 . evaluating point =0.2 of the evaluating point =0.2 plot software of the evaluating point =1.7 e-mail client of the semantic representation "a movie (that)", "an appreciation meeting (that)", "information (**)", and "creation (I want to carry out)" when the dictionary of drawing 2 is used to a word processor -- as -- the evaluating point for every software object is acquired, and a "word processor" with the highest evaluating point is chosen as optimal software object. After passing through the step which acquires acknowledgement from a user, you may make it choose a software object here, although a software object with the highest evaluating point may be chosen automatically.

[0012] Next, the environment for operating the software object chosen as mentioned above is set up

(step 53). Semantic representation is translated into a functional description expression using the dictionary (it is hereafter called a functional translation section dictionary) for specifically translating into the language which had the function of a software object normalized. An example of a functional translation section dictionary is shown in drawing 3. The functional translation section dictionary 50 of drawing 3 is the conversion table which made the input word and the output word (translation) which can be replaced one conversion pair. "Creation" which is for example, an input word is convertible for the output word of "making" with this conversion table. In the example of drawing 3, further, when there is a candidate of two or more output words to one input word, the evaluating point which shows the class of input word and the validity of conversion to an output word as additional information for choosing a suitable output word is given to each conversion pair. This evaluating point changes dynamically in process of an activity.

[0013] The conversion (translation) procedure from the input word based on the dictionary of drawing 3 to an output word is explained still more concretely. For example, if the word "creation" is seen, this language will be first translated for "making." Although it can translate "it makes" into either "a document is made", "a graphic form being made" and "e-mail being made" further, since the word processor is chosen as a software object here, the evaluating point of "making a document" is the highest. Therefore, "a document is made" is automatically chosen as a translation (translation) (or after acquiring acknowledgement from a user). "A document is made" is further translated for "newly drawing up the /document which starts a word processor." The entry which corresponds for "newly drawing up the /document which starts a word processor" does not exist in a dictionary. Therefore, it becomes the functional technical expression of "creation" "which newly draws up the /document which starts a word processor." similarly, if each word of the semantic representation "a movie (that)", "an appreciation meeting (that)", "information (**)", and "creation (I want to carry out)" is translated recursively, finally functional description expression of "starting a word processor", "guidance of a movie appreciation meeting", and "newly drawing up a document" up will be obtained.

[0014] Next, the operating instructions of a software object are generated from the above-mentioned functional description expression, and the instruction is executed (step 54). For example, the instruction description train for reading and performing the program of the word processor saved in the predetermined location of a hard disk to the functional description expression of "starting a word processor" is generated, and OS is performed. Moreover, to the functional description expression of "newly drawing up a document", the instruction description train for calling a document new creation function is generated, and the program of a word processor is performed through OS. The instruction description train which should be passed to OS is created according to the specification of the application programming interface (API) of OS, and the instruction description train which should be passed to the program of a word processor is created according to the specification of API of a word processor. The script for using various functions the command line for starting a program and inside a working program environment as an example of an instruction description train is mentioned.

[0015] Next, the result obtained by performing an instruction description train by OS or the software object is outputted with the predetermined gestalt which a user can recognize. For example, when the instruction which corresponds for "starting a word processor" is able to perform normally, the window of a word processor is displayed on a forefront side on the screen of a computer (step 55). Moreover, when the instruction which corresponds for "newly drawing up a document" is able to perform normally, the document of a blank paper is generated in the window of a word processor. In addition, when actuation is not able to perform normally, predetermined error processing is performed (step 56).

[0016]

[Effect of the Invention] As mentioned above, this invention chooses automatically the optimal software object for performing processing according to the demand of a user inputted with natural language, and offers the fundamental architecture which generates automatically the suitable instruction description train for operating the software object further. Cooperation with a software object and a natural language interface is more easily [than before] realizable with such this invention. That is, while defining the instruction description train for operating a software object, the structure which operates the software

object with natural language is realizable only by creating the dictionary data which associate each instruction description train and a functional description expression.

[0017] Moreover, to catching the character string of the inputted natural language with an instruction (namely, thing equivalent to the functional description expression of this invention) of a user, by this invention, it catches with the expression of a user's demand of the character string of the inputted natural language, a character string is analyzed, and the functional description expression of a software object is generated in-between at a Prior art using various dictionaries. In other words, a user is made to express what he wants to carry out by means of language by this invention to having made the user express how he wants to use the function of a software object conventionally by means of language. Therefore, even if a user does not know in advance what kind of software object can be used or with what kind of function each software object is equipped, he only expresses in direct language what he wants to carry out, and can operate a software object.

[0018]

[Example] The outline configuration of an example of computer system equipped with the natural language interface constituted according to this invention is shown in drawing 4 . This computer system was constituted using the common personal computer, and is equipped with the network controller 15, the user interface adapter 16, the display controller 21, and display 22 for cooperating with CPU (arithmetic and program control)10, ROM (read-only main storage)11, RAM (read-and-write possible main storage)12, the external storage controller 13, and external storage (auxiliary storage unit) 14 and the exterior. The various input devices (a keyboard 17, the microphone 18 for voice input, a mouse 19, tablet 20 for a handwriting input) for inputting a language train are connected to the user interface adapter 16.

[0019] The functional configuration of the system of this example is shown in drawing 5 . In drawing 5 , the natural language input section 30 is a means to generate the character string which receives the input of the language by natural language, a language train, or a sentence (these are hereafter called "language" collectively), and expresses the language. It can choose from actuation of the alphabetic character input panel on the screen using the key input using the keyboard 17 as the input approach of language, the voice input using a microphone 18, and a mouse 19, or the handwriting input using a tablet 20. of course -- approaches other than these -- the input device and the software of a sake for it (driver) - - even preparing -- if it carries out, it can use for the input of language.

[0020] The natural language analysis section 34 has natural language analysis, parsing in a dictionary, dialogic-operation functor generation, and a category dictionary manager function, analyzes the above-mentioned character string and generates semantic representation. The technique generally known for the field of natural language processing can be used for the analysis of a character string. For example, although natural language analytic engines of the "bamboo whisk" of Nara Institute of Science and Technology and Kyoto University, such as "KNP", are already known, the natural language analysis section 34 can be constituted using such an existing engine.

[0021] The configuration section 36 searches the configuration section dictionary 39 (refer to drawing 2) with all the concepts that appear in semantic representation, chooses the optimal software object for performing processing according to a demand of a user, and sets up the operating environment of the software object. In the configuration section dictionary 39, the information which associates an available software object, and the information about the configuration approach of a software object are saved in the concept and system which are used for semantic representation. A setup of the dictionary used for subsequent processings and a setup of the environment in the device by which a software object operates are included in configuration. When the configuration approach is described with natural language, a series of actuation is recursively repeated through the natural language analysis section 34.

[0022] The functional translation section 37 searches the functional translation section dictionary 40 (refer to drawing 3) with all the concepts that appear in semantic representation, and transposes it to the functional description expression suitable for the function of the software object memorized by the dictionary. Since the natural language itself may be registered into the dictionary, this permutation processing turns into recursive processing through the natural language analysis section 34. The

functional description expression finally generated turns into semantic representation which consists of the normalized language. In addition, when the undefined part is defined in the dictionary, the functional translation section 37 receives assignment of an undefined part from a user through the user dialogue section 31.

[0023] The instruction transfer section 38 searches the instruction transfer section dictionary 41 with all the concepts that appear in the functional description expression created by the functional translation section 37, and generates the instruction description train for performing the function of the software object 42 memorized by the dictionary. An instruction description train is series of commands passed through API, parameter, or command stream of the applicable software object 42. The instruction transfer section 38 performs an instruction description train, and performs the function of the applicable software object 42.

[0024] The response generating section 33 answers with the gestalt in which reception and a user ask for the activation result of the software object 42 performed by the instruction transfer section 38. The gestalt of a response can consider various things, such as a display on a display 22, printing by the printer (not shown), information storing in a database, and control of a device. The result obtained by performing the function of the software object 42 is inadequate, and when it cannot answer with the gestalt for which the user asked, the response generating section 33 shows a user a message through the user dialogue section 31, and if there is need, it will ask a user for directions.

[0025] The dictionary Management Department 35 performs creation of the new information on the configuration section dictionary 39, the functional translation section dictionary 40, and the instruction transfer section dictionary 41, modification of storing information, deletion, and a perusal display. Furthermore, a control section 42 delivers data required for the natural language input section 30, the natural language analysis section 34, the configuration section 36, the functional translation section 37, the instruction transfer section 38, the response generating section 33, the user dialogue section 31, and the dictionary Management Department 35, and controls a series of actuation.

[0026] It explains referring to drawing 1 - drawing 3 about the procedure of processing the input string of "wanting to create information of a movie appreciation meeting" by the system of this example.

[0027] If the user who will create information of a movie appreciation meeting and who thought drives in the sentence "want to create information of a movie appreciation meeting" using a keyboard 24, the natural language input section 30 will receive the character string of "wanting to create information of a movie appreciation meeting" through a keyboard entry interface (step 50). This character string is passed to the natural language analysis section 34.

[0028] the natural language analysis section 34 generates the semantic representation which analyzes the received character string, for example, consists of the functor and four words separated semantically "a movie (that)", "an appreciation meeting (that)", "information (**)", and "creation (I want to carry out)" (step 51). This semantic representation is passed to the configuration section 36.

[0029] The configuration section 36 acquires the evaluating point classified by software object of the four above-mentioned language based on the configuration section dictionary 39 (refer to drawing 2), and the highest software object of a synthetic evaluating point judges it to be a "word processor", and it performs configuration processing to a "word processor" word processor [which is stored in the configuration section dictionary 39] Turn (step 52). A setup of the functional translation section dictionary 40 and the instruction transfer section dictionary 41, the check of a computer resource, and reservation are included in configuration processing.

[0030] The functional translation section 37 translates semantic representation into a functional description expression by replacing in the function in which a software object offers the four above-mentioned language based on the functional translation section dictionary 40 (refer to drawing 3), and its combination (step 53). For example, although there is a candidate of two output words (translation), "a graphic form is made", about "it makes", since the evaluating point of "making a document" is the highest here, it changes for "making a document". ["a document is made" and] in this way, "the movie (that)" on which the natural language interpretation section 34 generated the functional translation section 37 -- the functional translation section dictionary 40 of drawing 3 searches and permutes

recursively the semantic representation "an appreciation meeting (that)", "information (**)", and "creation (I want to carry out)", and "a word processor is started", "guidance of a movie appreciation meeting", and the functional description expression of "newly drawing up a document" are generated. At the time of recursive retrieval and a permutation, semantic representation is dynamically changed using the natural language analysis section 34.

[0031] Next, the instruction transfer section 38 generates an instruction description train using the instruction transfer section dictionary 41 (step 54). if it takes for the example "which starts a word processor", first, the natural language analysis section 34 will analyze this character string, and will separate "a word processor (**)" and "starting (it carries out)." Next, the instruction transfer section 38 searches the instruction transfer section dictionary 41 with these concepts, and generates an instruction description train. In this case, "starting (it carries out)" replaces the software which starts the specific application for word processors through API which an operating system offers and which can be performed, and the instruction transfer section 38 performs. Generation of an instruction description train also performs retrieval and a permutation recursively, and changes semantic representation dynamically using the natural language analysis section 34.

[0032] Next, the response generation section 33 checks what the word processor started, and arranges a word processor to the forefront side of a display (step 55). In addition, when a word processor does not start according to a certain failure, the response generation section 33 converses with a user through the user dialogue section 31, and determines a solution (step 56). If a word processor starts, a user will create a document by carrying out the sequential input of the thing (demand) which he wants to do succeeding by means of language. For example, the language inputted says "a title is made information of a movie appreciation meeting", and "I want to emphasize a title." In addition, an input of the word of "ending" terminates a program.

[0033] In said example, "information of a movie appreciation meeting" was created because a user performs a series of natural language inputs. Next, by registering into a system the operating procedure which creates this "information" explains the procedure which enables it to re-create same "information" simply. Here, it thinks of enabling it to create as an example "information" which changed introduction of time, a location, a movie name, and a movie freely.

[0034] First, the functional description expression equivalent to a series of actuation in which it explained previously is suitably registered into the functional translation section dictionary 40 by the identifier (here, it considers as "information of a movie appreciation meeting") through the dictionary Management Department 35. Next, time, a location, a movie name, and the part equivalent to introduction of a movie are redefined as an undefined part through the dictionary Management Department 35 among the character strings contained in said a series of functional description expressions registered into the functional translation section dictionary 40 (step 60).

[0035] Next, through the dictionary Management Department 35, the character string "information of a movie appreciation meeting" is related with a word processor software object, and it registers with the configuration section dictionary 39 (step 61).

[0036] If a user performs a natural language input, saying "information of a movie appreciation meeting being created" through the natural language input section 30 after adding the entry of "information of a movie appreciation meeting" to the functional translation section dictionary 40 and the configuration section dictionary 39 as mentioned above, the natural language analysis section 34 and the configuration section 36 will perform the same processing as the example of drawing 1, and will generate semantic representation (step 62).

[0037] Next, in case the functional translation section 37 translates said semantic representation into a functional description expression, it transposes to a series of functional description expressions which registered "information of a movie appreciation meeting" into the functional translation section dictionary 40 previously (step 63), and performs recursive translation processing which was previously explained to the functional description expression (step 64). If the undefined part (introduction of time, a location, a movie name, and a movie) contained in a functional description expression is found (step 65), the functional translation section 37 will ask a user about the definition of the part through the user

dialogue section 31. If a user inputs the language (character string) equivalent to a definition, the functional translation section 37 will replace an undefined part in the language (step 66). Thus, along with guidance of the user dialogue section 31, a user is inputting introduction of time, a location, a movie name, and a movie, and can create information of a movie appreciation meeting easily.

[0038] As mentioned above, although the example of this invention was explained, an example is not restricted above. For example, although two or more application software objects installed in the personal computer shall be operated through a natural language interface in the above-mentioned example, it is also possible to enable it to operate two or more electronic equipment (for a computer to also be included) corresponding to a network connected to networks, such as LAN and the Internet, through the natural language interface of a control equipment connected to this network. The system of forming one voice input mold controller to the mold electric appliances corresponding to a network which followed, for example, were connected to domestic LAN is also realizable.

[Translation done.]

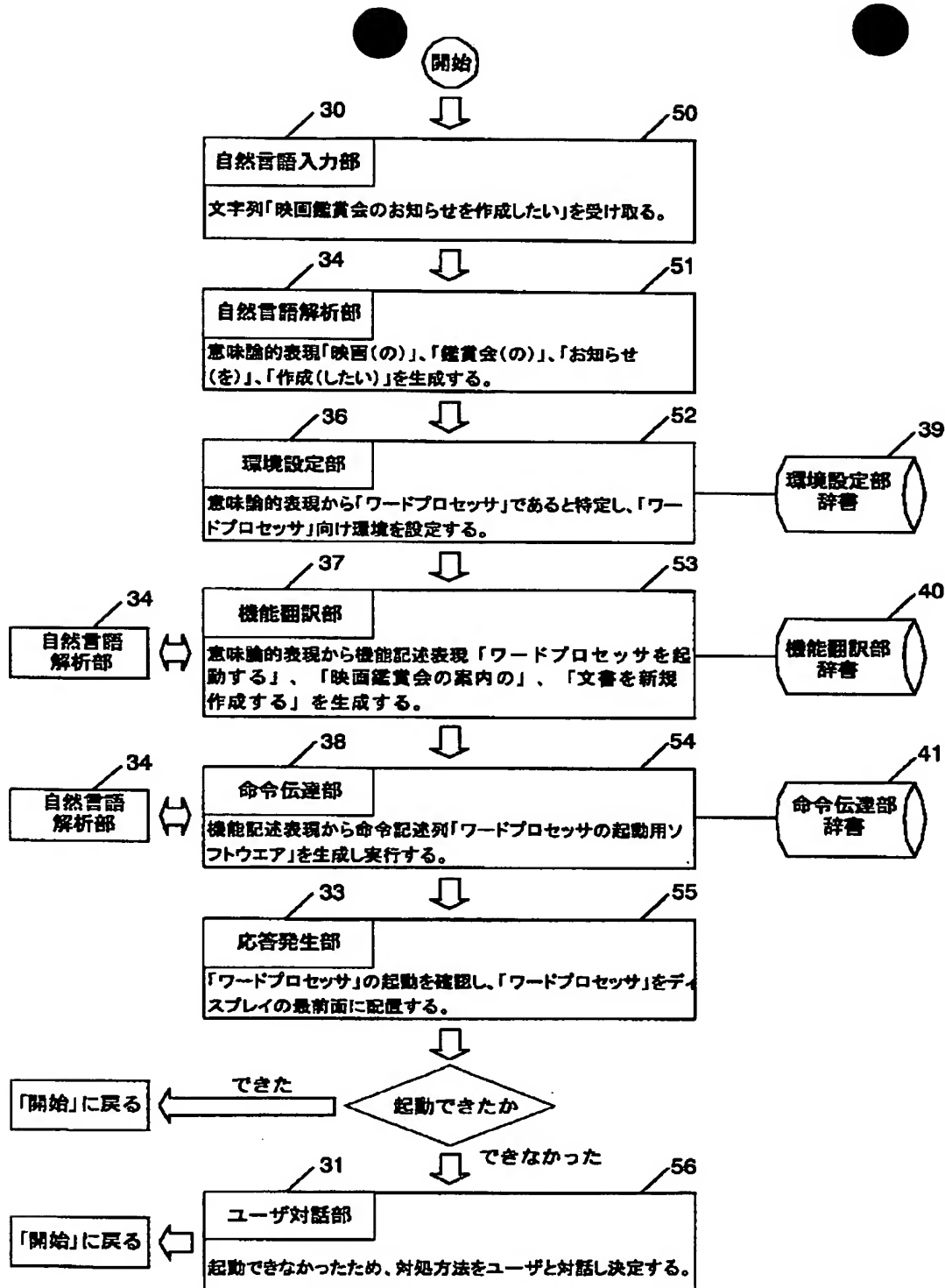
*** NOTICES ***

JPO and NCIPI are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DRAWINGS

[Drawing 1]



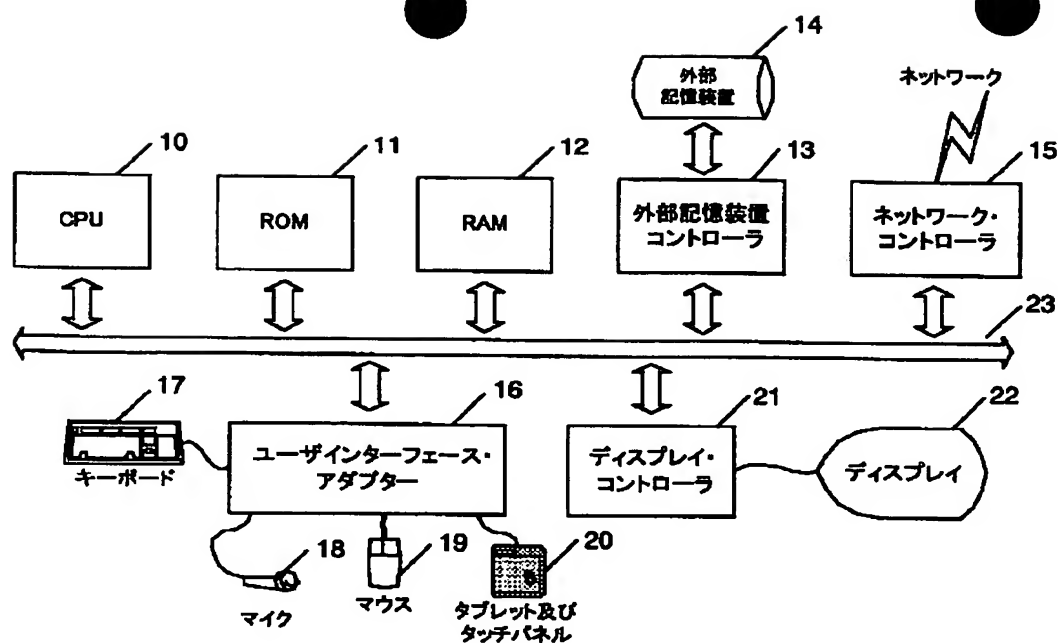
[Drawing 2]

入力語	種類	評価点	出力語
お知らせ	名詞	1	ワードプロセッサ
お知らせ	名詞	1	メール
案内	名詞	0.8	ワードプロセッサ
案内	名詞	0.4	メール
作成	動詞	0.7	ワードプロセッサ
作成	動詞	0.1	図形作成ツール
作成	動詞	0.2	メール
送る	動詞	1	メール

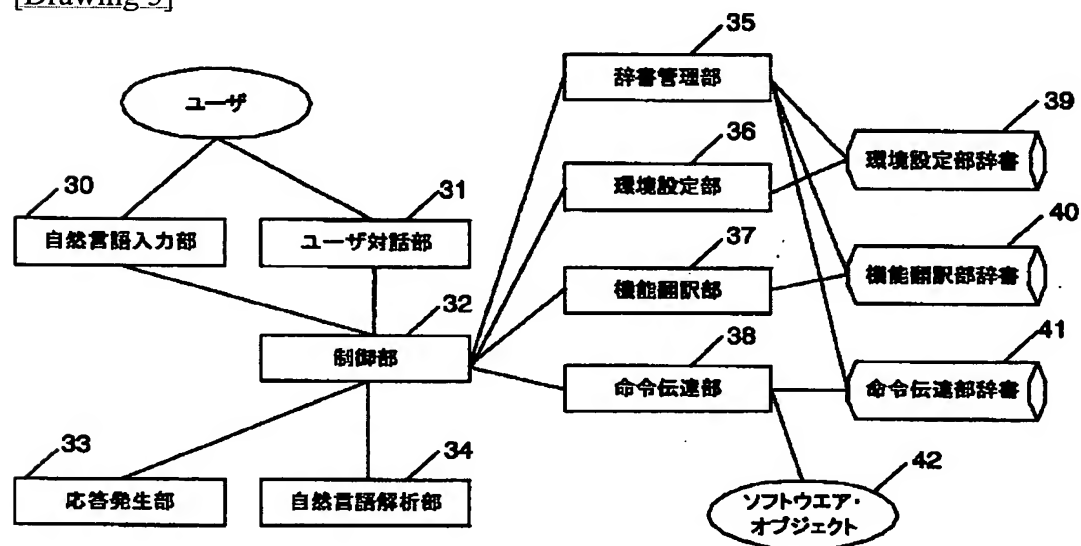
[Drawing 3]

入力語	種類	評価点	出力語
作成	動詞	1	作る
製作	動詞	1	作る
お知らせ	名詞	1	案内
ご案内	名詞	1	案内
案内	名詞	0.6	案内の文書
案内	名詞	0.4	案内のメール
作る	動詞	0.7	文書を作る
作る	動詞	0.1	図形を作る
作る	動詞	0.2	メールを作る
送る	動詞	1	メールを送る
文書を作る	動詞	1	ワードプロセッサを起動する/文書を新規作成する
図形を作る	動詞	1	図形作成ツールを起動する/図形を新規作成する
メールを作る	動詞	1	メールツールを起動する/メールを新規作成する
メールを送る	動詞	1	メールを作る/メールを送信する

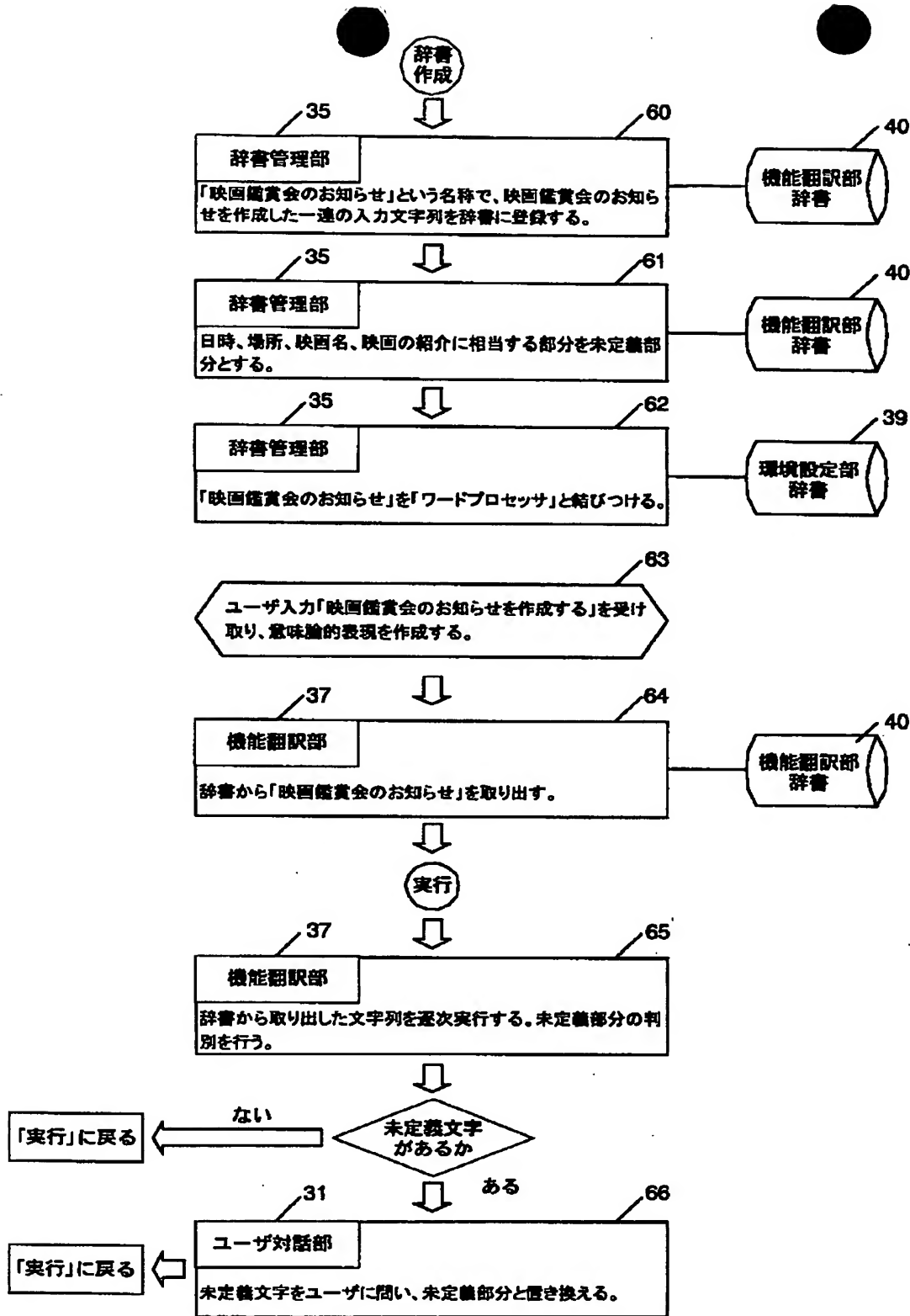
[Drawing 4]



[Drawing 5]



[Drawing 6]



[Translation done.]

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☒ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☒ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.